REMARKS

Claims 1-8, 11-12 and 17-29 are pending in this application. By this Amendment, the specification and claims 1-5 and 12 are amended and new claims 17-29 are added. Attached hereto is a marked-up version of the changes made to the specification and claims by the current amendment. The attached page is captioned "Version With Markings To Show Changes Made."

The Office Action indicates that Figure 13 should be designated as prior art. The attached Request for Approval of Drawing Corrections appropriately labels Figure 13 as prior art.

The Office Action objects to the disclosure because of informalities. The above-amendments to the specification as well as various amendments to the claims are intended to correct typographical and/or translational errors in the original application. These amendments are made for clarity and not for reasons of patentability. Withdrawal of the objection to the disclosure is respectfully requested.

The Office Action objects to the claims because of informalities and rejects claim 5 under 35 U.S.C. § 112, second paragraph. It is respectfully submitted that the above amendments to the claims obviate the grounds for objection and rejection. That is, the claims have been amended for clarity and claim 5 is amended to remove the disputed terminology. Withdrawal of the objection and rejection are respectfully requested.

The Office Action rejects claims 1, 5, 8 and 11 under 35 U.S.C. § 102(e) over U.S. Patent 5,752,272 to Tanabe. The Office Action also rejects the various dependent claims under various combinations of Tanabe in view of U.S. Patent 5,778,422 to Genduso, U.S. Patent 5,935,253 to Conary et al., U.S. Patent 5,829,031 to Lynch, The Cache Memory Book by Handy, U.S. Patent 5,381,532 to Suzuki and U.S. Patent 5,357,618 to Mirza. The rejections are respectfully traversed with respect to the pending claims.

Independent claim 1 recites a system bus connecting the processor and the memory controller, and at least two memory buses connecting the memory controller and the memory. The at least two memory buses include a first memory bus for transferring instruction code and a second memory bus for transferring data. Additionally, independent claim 1 recites that the control circuit estimates a most probable address to be accessed next in the memory, and that the access judging circuit prefetches data stored in the most probable address of the memory into the buffer.

Tanabe does not teach or suggest all the features of independent claim 1. That is, Tanabe merely discloses a micro-processor 3, a memory access control device 1 and a Rambus DRAM 5. As shown in Figure 4, the Rambus interface 33 is connected to the Rambus DRAM 5 by a Rambus line RB. See also Tanabe's column 6, lines 39-40. Tanabe does not teach or suggest a first memory bus for transferring an instruction code and a second memory bus for transferring data in combination with a control circuit estimating a most probable address to be

accessed next in the memory, and that the access judging circuit prefetches data stored in the most probable address of the memory into the buffer. Tanabe merely shows the Rambus line RB between the DRAM 5 and the Rambus interface 33. Tanabe does not suggest two memory buses for transferring instruction code on one memory bus and for transferring data on another memory bus.

Tanabe also does not teach or suggest the advantages of the claimed features, namely that since the instruction code memory is separated from the data memory and the memory bus and its control circuit are provided for each of the memories, then contention may be avoided on the memory bus between the instruction code read ahead and the data access. See on page 27, line 26-page 28, line 3 of the present specification.

Each of the other cited references, either alone or in combination, also do not teach or suggest the respective features of independent claim 1. Furthermore, there is no suggestion of how to modify Tanabe with any of the other cited references to find the claimed features of claim 1. That is, there is no suggestion to modify Tanabe's system to include memory, a first memory bus for transferring an instruction code, a second memory bus for transferring data in combination with the control circuit and the access judging circuit as recited in claim 1. There is no suggestion in the prior art to modify Tanabe's structure to include multiple memory buses for transferring instructions and data. Any such modification would destroy the express purpose of Tanabe. Any such combination is therefore impermissible. The Office Action also relies on impermissible hindsight for any such modification of

Tanabe since there is no suggestion in the prior art to include the claimed memory buses, control circuit and access judging circuit. Accordingly, independent claim 1 defines patentable subject matter.

Independent claim 18 also defines patentable subject matter for at least similar reasons as claim 1 by reciting a first memory bus connecting the memory controller and the memory, a second memory bus connecting the memory controller and the memory as well as the memory controller including a control circuit to estimate a most probable address to be accessed next in the memory and an access judging circuit to prefetch data stored in the most probable address of the memory.

Each of claims 2-8, 11-12, 17 and 19-29 depend from one of the independent claims and therefore also defines patentable subject matter.

In addition, the dependent claims also recite features which further and independently distinguish over the prior art. For example, the cited references do not teach or suggest that the access judging circuit prefetches the instruction code from the memory along the first memory bus and into the buffer as recited in dependent claims 17 and 28.

For at least the reasons set forth above, it is respectfully submitted that claims 1-8, 11-12 and 17-29 define patentable subject matter. Withdrawal of the outstanding rejections is respectfully submitted.

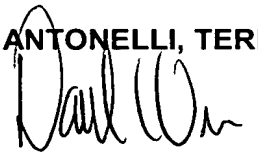
CONCLUSION

In view of the foregoing, it is respectfully submitted that the above- identified application is in condition for allowance. Favorable consideration and prompt allowance of claims 1-8, 11-12 and 17-29 are respectfully requested.

Please charge any shortage in fees due in connection with the filing of this paper, including extension of time fees, or credit any overpayment of fees, to the deposit account of Antonelli, Terry, Stout & Kraus, LLP, Deposit Account No. 01-2135 (500.36683CX1).

Respectfully submitted,

ANTONELLI, TERRY, STOUT & KRAUS, LLP



David C. Oren
Registration No. 38,694

DCO/sjl
(703) 312-6600

VERSION WITH MARKINGS TO SHOW CHANGES MADE

IN THE SPECIFICATION:

The paragraph beginning on page 1, line 9 has been amended as follows:

Fig. 13 shows an arrangement of a general information processing system as a prior art. A processor 1 and a memory controller 2 are connected by a system bus 110, the memory controller 2 and a memory 3 are connected by a memory bus 111, and the memory controller 2 and another system are connected by an IO bus (not shown). The processor 1 of the present system includes an on-chip cache (which will be referred to as the L1 cache, hereinafter) 12, and an L2 cache 14 connected to the system bus 110. The memory controller 2 performs connection control not only over the memory 3 and L2 cache 14 but also over the other system. The operation of the processor 1 of reading an instruction code (which operation will be referred to as fetch, hereinafter) is summarized as follows. The processor 1 issues a memory access request to the memory controller 2 via the instruction processing part 11 and system bus 110. The memory controller 2, in response to the request, reads an instruction code from the L2 cache 14 or memory 3 and transmits it to the processor 1. An access size between the processor 1 and memory 3 is influenced by the L1 cache 12 so that the reading of the code from the memory 3 is carried out on every line size basis as the management

unit of the L1 cache 12. Most processors are [each] typically equipped [usually] with[, in addition to an L1 cache,] an L2 cache provided outside the processor core as a relatively high-speed memory in addition to an L1 cache. The word 'cache'as used herein refers to a memory which stores therein an instruction code once accessed by a memory to realize a high-speed access to the same code in the case of an occurrence of the re-access to the same code. In order to perform arithmetic operation, the processor also makes access not only to such an instruction code but also to various sorts of data including operands and to external registers. Even these data is stored in an cache in some cases. Such a technique is already implemented in many systems including a personal computer as a typical example.

The paragraph beginning on page 2, line 23 has been amended as follows:

In an information processing system, in addition to the arithmetic operation performance of a processor, the reading performance of an instruction code from a memory to the processor is also important. A delay from the access request of the processor to the acceptance of the data thereof is known as [an] access latency. In these years, the core performance of the processor has been remarkably improved, but an improvement in the supply capability of the instruction code from the access memory is still insufficient. When the access latency becomes unnegligible due to a performance

difference between the both, the operation of the processor stalls, which disadvantageously results in that the processor cannot fully exhibit the performances and thus the memory system becomes a bottleneck in the system. Such [an] access latency [problem occurs] problems occur not only for the instruction fetch but also for data or register operands.

The paragraph beginning on page 3, line 14 has been amended as follows:

Conventional methods for improving [an] access latency include first to fourth methods [which follow] as will be described.

The paragraph beginning on page 3, line 25 has been amended as follows:

The second improvement method is to speed up the memory. [For the speed-up of] In order to speed up the memory, [it is considered to] one may speed up the operation of the memory per se and also [to] use a cache as the memory. However, such a high-speed memory as a high-speed SRAM or a processor-exclusive memory is expensive, which undesirably involves an increase in the cost of the entire system. Meanwhile the cache has problems based on its principle as follows. That is, the cache is effective after once accessed and is highly useful when repetitively accessed. In particular, a program to be executed on a so-called embedded processor tends to have a low locality of references, the re-use frequency

of an instruction code is low and thus the cache memory cannot work effectively. This causes the instruction code to have to be read out directly from the memory, for which reason this method cannot make the most of the high-speed feature of the cache. Further, such a high-speed cache memory used as a high-speed SRAM or a processor-exclusive memory is expensive. Though the price/performance ratio of the memory is improved, the employment of the latest high-speed memory involves high costs. An increasingly large capacity of memory has been demanded by the system in these years. Thus the cost increase becomes a serious problem.

The paragraph beginning on page 9, line 27 has been amended as follows:

Fig. 1 is a general block diagram of an embodiment of the present invention. The present embodiment is an example wherein a memory 3 stores therein an instruction code to be executed on a processor 1 and data such as operands to perform [prefetchomg] a prefetching operation for an instruction code access.

The paragraph beginning on page 11, line 8 has been amended as follows:

The memory controller 2 will [then] be [detailed] described in detail below.

The paragraph beginning on page 11, line 23 has been amended as follows:

The control circuit 5 performs control over the entire memory controller. [More in detail, the] The control circuit 5 also performs read-ahead control from the instruction code memory 32, in addition to control over the switch circuits 6, 9, memory control circuits 21, 22, system bus control circuit 20, etc. Details of the control circuit 5 will be explained in connection with Figs. 4, 8 and 9.

The paragraph beginning on page 13, line 25 has been amended as follows:

Explanation will [next] now be made as to an implementation example of the access judgement circuit. [Shown in] Fig. 2 is a block diagram of an example of the access judgement circuit 4 in the memory controller 2 of Fig. 1 in the present invention. The access judgement circuit 4 has a prefetch hit judgement circuit 41 and an instruction fetch detection circuit 42. The prefetch hit judgement circuit 41 has a prefetch address register 411 for storing therein the address of the prefetched instruction code and a comparator 412 for comparing the address accessed by the processor with the address prefetched by the memory controller. When [the] both addresses [are coincided] coincide

with each other, the prefetch hit judgement circuit 41 [can] judges it as a prefetch hit. The instruction fetch detection circuit 42 has an instruction-code memory area address register 421 for storing therein an upper address indicative of the instruction code memory area and a comparator 422 for comparing the upper address of the instruction-code memory area address register 421 with the address accessed by the processor.

The paragraph beginning on page 14, line 18 has been amended as follows:

[Though] Although not illustrated in FIG. 2, the access judgement circuit further includes an access read/write judgement circuit. When a coincidence is found in the comparison and the access is of a read type, the judgement circuit can determine it as an instruction code fetch. For example, in the case where the instruction code memory area is from 100 00000H to 10FF FFFFH, 10H as upper 8 bits of the upper address is previously set in the instruction-code memory area address register 421, an access to the instruction code area can be detected from the comparison result of the upper 8 bits of the address accessed by the processor. The setting of the instruction-code memory area address register 421 is required only once at the time of the initialization setting.

The paragraph beginning on page 15, line 5 has been amended as follows:

As [has been mentioned] described above, the present embodiment [is featured in] provides that detection of the instruction code fetch is carried out by judging whether or not the access address of the processor is placed in the instruction code memory area, the detection of the fetch access of the instruction code and the prefetch hit judgement are carried out at the same time, whereby access judging operation can be realized with a small overhead time.

The paragraph beginning on page 15, line 27 has been amended as follows:

[Explanation] An explanation will [then be made] now be provided as to a control circuit for performing read-ahead control, transfer control over the processor, and control over the entire memory controller. Fig. 4 is a block diagram of an example of the control circuit 5 in the memory controller in the present invention of Fig. 1. The control circuit 5 includes a prefetch address generation circuit 51, a prefetch sequencer 52 and a selector 53.

The paragraph beginning on page 16, line 20 has been amended as follows:

The subject matter of this method is to calculate an address to be [next accessed] accessed next thus not to restrict the access size to the line size of the level-i cache. Further, the line size value 511 may be a fixed value or a variable value by a register. The prefetch sequencer 52, on the basis of information received from the system bus control line or access judgement circuit 4, executes a memory access and a prefetch from the memory according to the access of the processor.

The paragraph beginning on page 17, line 6 has been amended as follows:

[Referring to] Fig 5[, there is shown] is a block diagram of an example of the buffer memory 8 in the memory controller [in] of the present invention. In some processors, it is impossible to read addresses sequentially from its smaller address in a burst read access of level-1 cache filling operation. This is because the most critical instruction code is read ahead. For example, when it is desired to read 32-bit data having continuous addresses 0, 1, 2 and 3; the data may not be read in the address ascending order of 0, 1, 2 and 3 but may be read in an address order of 2, 3, 0 and 1. In order to solve such an access problem, in the present example, the buffer memory 8 [was made up of] includes a plurality of buffer memories having a width equal to the access size of the processor. More specifically, in the example, an instruction code is assumed to consist of 32 bits, 4 channels of buffer memories 0 to 3 each having a 32-bit width are provided so that data are stored in the buffer memories sequentially from the buffer memory 0 at the time of reading from a

memory, whereas, data transfer is carried out in an order requested by the processor in the processor transfer mode. As a result, the present invention can flexibly be compatible with any processor access system.

The paragraph beginning on page 18, line 2, has been amended as follows:

[Shown in] Fig. 6 is a block diagram of another embodiment of the memory controller [in] of the present invention. The present embodiment [is featured in] provides that the memory controller 2 [newly] includes an instruction decoder circuit 43 for decoding and analyzing an instruction code transferred from the instruction code memory 32 to the memory controller 2 and also includes a branching buffer memory 84. The instruction decoder circuit 43 detects presence or absence of a branch instruction such as branch or jump in the transferred instruction code. The control circuit 5, when the instruction decoder circuit 43 detects a branch instruction, reads ahead an instruction code at the branch destination into the branching buffer memory 84. The access judgement circuit 4, in the presence of an instruction code access from the processor, judges whether or not it is found in the normal read-ahead buffer memory 8 or in the branching buffer memory 84. In the case of a hit, the control circuit 5 transfers the instruction code from the hit buffer memory to the processor. As a result, even when a branch takes place in the processor, performance deterioration caused by stall can be improved.

The paragraph beginning on page 18, line 25, has been amended as follows:

Fig. 7 is a block diagram of another embodiment of the memory controller [in] of the present invention. The present embodiment [is featured in] provides that a buffer memory and a control circuit are [provided] includes not only for the instruction code area but also for the data memory area and register area, individually.

The paragraph beginning on page 19, line 3 has been amended as follows:

An access from the processor is divided by the switch circuit 90 into accesses to the instruction code area, data area and register areas. The access judgment circuit 4 judges a hit in each buffer memory. The access judgement circuit 4 can be easily implemented in substantially the same manner as in [that in] the embodiment of Figs. 3 and 4. The control circuit 5 has a data access control circuit 501, an instruction code access circuit 502 and an I/O control circuit 503. Each control circuit has a sequencer for prefetch control to implement a prefetch for each area. Further, even switch circuits 61, 62, 63, direct paths 71, 72, 73 and buffer memories 81, 82, 83 are provided for each area.

The paragraph beginning on page 20, line 8 has been amended as follows:

Next, an explanation will be [made as to] provided of the operation of the prefetch sequencer 52 by referring to Figs. 8 and 9. Fig. 8 shows a flowchart of an exemplary operation of the prefetch sequencer 52 in Fig. 4. This exemplary

flowchart shows when data corresponding to one access size is prefetched from an address following the current access for preparation of the next access at the time of occurrence of the access to the instruction code area.

The paragraph beginning on page 20, line 17 has been amended as follows:

When a processor access takes place, the prefetch sequencer 52 first judges whether or not this access is a read access to the instruction code area (step 201). The judgement is implemented, e.g., by means of address comparison, and its comparison circuit is implemented with the access judgement circuit 4. In the case of the read access to the instruction code area, the sequencer judges whether or not a prefetch hit occurs (step 202). Even for this judgement, a judgement result of the access judgement circuit 4 is used. In the case of a hit, the sequencer starts data transfer from the buffer within the memory controller to the processor (step 203). In the case of no hit, the sequencer performs the data transfer from the memory to the processor via the direct path (step 204). Further, since the data within the prefetch buffer is not a prefetch hit data, the prefetch buffer is cleared (step 205).

The paragraph beginning on page 22, line 13 has been amended as follows:

This embodiment [is featured in] provides that continuous instruction codes estimated to be [next accessed] accessed next are fetched until the buffer becomes full of the codes to reach its full storage capacity (buffer full). In this conjunction, it is desirable to set the buffer capacity to be an integer multiple of the access size. As a

result, since the transfer between the memory and the buffer memory of the memory controller can be carried out with a relatively long burst size at a time, the need for performing the read-ahead operation for each instruction code access from the processor can be eliminated and control can be facilitated.

The paragraph beginning on page 22, line 25 has been amended as follows:

Fig. 10 is a timing chart showing an exemplary memory access [in] of the present invention. In this example, the prefetch effect at the time of the memory access will be explained by comparing it with that of the prior art. It is assumed herein as an example that the processor reads an instruction code through two burst read accesses for each cache line size on the basis of continuous addresses of from 0000 to 001F. Four words of '0000' in the first access and 4 words of '0010' are burst-read respectively in 4 cycles.

The paragraph beginning on page 28, line 4 has been amended as follows:

The read-ahead to the memory controller is carried out at the time of a processor access so that, at the time of a read-ahead hit, the data of the buffer memory is transferred to the processor and at the same time, an address to be [next accessed] accessed next by the processor is estimated to perform the read-ahead operation from the buffer to the buffer memory. At the time of a read-ahead error, the data is transferred from the memory directly to the processor and at the same time, the data of the buffer memory is cleared, an address to be [next accessed]

accessed next by the processor is estimated to perform the read-ahead operation from the memory to the buffer memory. As a result, at the time of a read-ahead error, the read-ahead access can be realized simultaneously with the access to the processor, whereby the system can cope with continuous access requests from the processor.

IN THE CLAIMS

Claims 17-26 have been added.

Claims 1-5 and 12 have been amended as follows:

1. (Twice Amended) An information processing system comprising:

a processor;

a memory; [and]

a memory controller [connected with said processor via a first bus and connected with said memory via a second bus for controlling said memory];

a system bus connecting said processor and said memory controller; and;

at least two memory buses connecting said memory controller and said memory, said at least two memory buses comprising:

a first memory bus for transferring an instruction code, and

a second memory bus for transferring data,

said memory controller [further] comprising:

a buffer,

a control circuit, and

an access judging circuit, wherein;

said control circuit estimates [an] a most probable address to be accessed next in said memory, and

said access [judgement] judging circuit prefetches [a] data stored in said most probable address of the memory into the buffer [memory, before a memory access is carried out from said processor].

2. (Twice Amended) An information processing system according to claim 1, wherein said memory controller comprises a direct path for transmitting data directly to said processor from said memory therethrough[; said control circuit, when the access from said processor hits data within said buffer, is controlled to transfer the data to said processor, whereas, said control circuit, when the access from said processor fails to hit data within said controlled to transfer data within said memory to said processor via said direct path.

3. (Twice Amended) An information processing system according to claim 1, wherein said memory stores [an] said instruction code to be executed on said processor therein, and said control circuit prefetches the instruction code into said buffer.

4. (Twice Amended) An information processing system according to claim 1, wherein said memory stores therein [an] said instruction code to be executed on said processor and operand data, and said control circuit prefetches the instruction code and said operand data into said buffer.

5. (Twice Amended) An information processing system according to claim 1, wherein said memory controller further comprising a plurality of buffers, [into which data of said access unit is prefetched, and] wherein said control circuit [controls to transfer] transfers data already stored in said plurality of buffers to said processor in an order different from an address order.

12. (Amended) An information processing system according to claim 1, wherein said memory is divided into a first memory for storing therein [an] said instruction code to be executed on said processor and a second memory for storing therein operand data, [; said memory controller has an] wherein said access [judgement] judging circuit for judging whether the memory access from said processor is an access to said first memory or an access to said second memory, said memory controller including a first buffer memory for prefetching of the instruction code and a second memory for prefetching of the operand data; and said control circuit is controlled to prefetch the instruction code into said first buffer memory according to [an] a judgement of

said access [judgement] judging circuit or to prefetch the operand data into said second buffer memory.